贪心

在求解某些问题时，只需要做出在当前看来是最好的选择就能获得最好的结果，而不需要考虑整体上的最优，即使目光短浅也是没有关系的。

1. 贪心与证明

贪心算法不是对所有问题都能得到整体最优解，关键是贪心策略的选择，选择的贪心策略必须具备无后效性，即某个状态以前的过程不会影响以后的状态，只与当前状态有关。首先需要证明贪心策略是正确的，才可以考虑使用贪心算法解决该问题。在很多情况下，贪心的合理性并不是显然的，但如果能找到一个反例，就可以证明这样的贪心不正确。

**部分背包问题：**

藏宝洞里面有N(N<=100)堆金币，第i堆金币的总重量和总价值分别是mi和vi(1<=mi,vi<=100)。阿里巴巴有一个承重量为T(T<=1000)的背包，但并没办法将全部的金币都装进去。他想装走尽可能多价值的金币。所有金币都可以随意分割，分割完的金币重量价值比(也就是单位价格)不变。请问：阿里巴巴最多可以拿走多少价值的金币？

证明贪心的第一种方法：假设要选择的方案不是贪心算法所要求的方案，只需要证明将需要贪心的方案替换掉要选择方案，结果会更好(至少不会更差)。

如果藏宝洞里面不是一堆堆金币，而是一个个单价不一且无法分割的金块，通过贪心策略得到的却不是最优解。正确做法是**搜索(?)**或者**动态规划**。

**动态规划：**动态规划(Dynamic Programming，DP)是运筹学的一个分支，是求解决策过程最优化的过程。

**背包问题：**背包问题是一类经典的动态规划问题。

**分治：**分治字面上的解释是“分而治之”，就是把一个复杂的问题分成两个或更多的相同或相似的子问题，再把子问题分成更小的子问题……直到最后子问题可以简单的直接求解，原问题的解即子问题的解的合并。分治法是很多高效算法的基础，如排序算法（快速排序，归并排序）。

**凌乱的yyy：**

现在各大OJ上有n个模拟比赛，每个比赛的开始、结束的时间点(a,b)是知道的。yyy认为，参加数量越多的模拟比赛，正式比赛成绩就越好。如果要参加一个比赛必须善始善终，而且不能同时参加两个及以上的比赛。他想知道他最多能参加几个比赛。

根据分析，应该选择参加最先结束的那一场比赛。接下来，要选择能够参加的比赛中，最早结束的比赛(既然已经决定参加上一场比赛了，那么所有和上一场冲突的比赛都不能参加了)，直到没有比赛可以参加为止。这样可以保证不管在什么时间点之前，能够参加比赛的数量都是最多的，因此贪心算法成立。

这就是证明贪心的另外一种方法——数学归纳法：每一步的选择都是到当前为止的最优解，一直到最后一步就成为了全局的最优解。

本题中将所有比赛的结束时间排序，然后依次进行贪心：如果能够参加这场比赛，就报名参加；如果这场比赛和上一场冲突，就放弃。

1. 哈夫曼编码

算法中基本的算法思想有：贪心算法、分治算法、回溯算法、动态规划。这些算法思想并不是具体的算法，他们是用来指导我们设计具体的算法和编码的算法思想。贪心算法有很多经典的应用：哈夫曼编码、Prim和Kruskal最小生成树算法、Dijkstra单源最短路径算法。

**定义：**在权为w1,w2,…,wn的n个叶子结点的所有二叉树中，带权路径长度WPL最小的二叉树称为赫夫曼树或最优二叉树。

**哈夫曼树的构造(哈夫曼算法)**

1.根据给定的n个权值{w1,w2,…,wn}构成二叉树集合F={T1,T2,…,Tn},其中每棵二叉树Ti中只有一个带权为wi的根结点,其左右子树为空.

2.在F中选取两棵根结点权值最小的树作为左右子树构造一棵新的二叉树,且置新的二叉树的根结点的权值为左右子树根结点的权值之和.

3.在F中删除这两棵树,同时将新的二叉树加入F中.

4.重复2、3,直到F只含有一棵树为止.(得到哈夫曼树).

**哈夫曼编码**

哈夫曼树的应用很广，哈夫曼编码就是其在电讯通信中的应用之一。广泛地用于数据文件压缩的十分有效的编码方法。其压缩率通常在20%～90%之间。在电讯通信业务中，通常用二进制编码来表示字母或其他字符，并用这样的编码来表示字符序列。

**优先队列：**

**二叉堆:**

int a1[10010];  
memset(a1,127,sizeof(a1));//将数组初始化为一个接近int最大值的数，效率较高

使用memset初始化int数组时，第二个参数如果是0，数组就会被初始化为0；如果是127，会初始化为一个很大且接近int类型上限的正数；如果是128，会初始化成很小且接近int类型下限的负数；如果是-1或者255时，数组会初始化为-1.